

OPERACIÓN DIRIGIDA POR MODELO DE UNA CÉLULA DE MANUFACTURA FLEXIBLE CON CONTROLADOR LÓGICO PROGRAMABLE

José E. Acosta Cano de los Ríos¹, Adolfo Cancino Herrera, Pedro R. Acosta Cano de los Ríos, Óscar Chávez López, Víctor M. Ambriz Díaz, Israel Rosas Yescas
Tecnológico Nacional de México / Instituto Tecnológico de Chihuahua
División de Estudios de Posgrado e Investigación
Ave. Tecnológico No 2909
¹jose.ac@chihuahua.tecnm.mx

RESUMEN.

Se presenta la aplicación de la técnica de modelado iMRP en el control de una célula de manufactura flexible controlada por controlador lógico programable (PLC), tomando como base el concepto de operación dirigida por modelo. Partiendo del desarrollo de un programa principal (uso genérico) que interpreta el modelo iMRP del sistema particular (específico), se presenta una manera de controlar la secuencia de operación de un conjunto de equipos de fabricación programable (Robot y Torno). EN el modelo del sistema particular se abstrae la estructura del sistema de fabricación y la lógica a seguir para su funcionamiento (secuencia y operaciones a realizar). El proyecto se realizó con base en un robot y un centro de maquinada CNC, ambos de uso industrial.

Palabras Clave: Ejecución dirigida por modelo, FMS, control basado en PLC

ABSTRACT.

An application of the iMRP modeling technique in the control of a flexible manufacturing cell is presented. The cell is controlled by a programmable logic controller (PLC). A generic application is running in the PLC: This application interprets the iMRP model of the particular system (specific case) in order to control the operation sequence of a set of programmable manufacturing equipment (Robot and Lathe). The iMRP model abstracts the particularities of the system, that is, the structure of the manufacturing system and the logic to follow for its operation (sequence and operations to be carried out). The project was carried out based on a robot and a CNC machining center, both of them are industrial grade equipment.

Keywords: Model-driven-execution, FMS, PLC based control

INTRODUCTION

El modelado como apoyo a la programación existe desde inicios de la computadora electrónica. En la década de los cuarenta, antes de ser la ingeniería de software identificada como tal en 1968 [1], Goldstine y von Newman propusieron el uso de diagrama de flujo, [2]. El desarrollo basado en modelo (MDD) es una metodología caracterizada principalmente por el uso de modelos como productos, [3] [4], [5]. Actualmente en esta área el esquema MDA propuesto por [6], es uno de los esquemas más difundido y aceptado en donde se encuentra la transformación automatizada de modelo a código [7], [8], [9], [10], [11], [12].

EL uso del lenguaje UML en MDD se hace patente en extensiones ligeras de UML para su aplicación en un dominio específico, con lo que se conoce como perfiles de UML, [13], tales como, sysML [14] y [15], UML4IoT [16], UML4sPM [17], packML [18] y plcML [19], [20]. entre otros.

Otros lenguajes de modelado han sido desarrollados específicamente para el dominio de sistemas basados en PLC. En [21] se describe el método de representación formal PLCSpecif, con enfoque a la definición formal de la lógica de control para su transformación en código. Otros métodos clásicos en el dominio de PLC's son Grafcet, [22] (IEC-60848 2013) y la referencia GEMMA, [23].

Otra vertiente del uso del modelo es la operación dirigida por modelo en donde no existe una transformación (a código) del modelo, sino una relación de instanciación del modelo desarrollado, lo que algunos autores denominan progenitura [24]. El modelo es directamente interpretado por un sistema genérico para guiar la operación del sistema en una aplicación particular, sin transformar el modelo a código, [27]. En práctica actual de desarrollo de sistemas informáticos este concepto puede relacionarse con lo que se conoce como plataformas no code o low code, [27].

A pesar de los avances en el modelado de sistemas basados en PLC, la operación dirigida por modelo no es reportada en la literatura revisada en el dominio de sistemas basados en PLC. El objetivo del presente artículo es describir una manera de aplicar el concepto de operación dirigida por modelo en sistemas basados en PLC. Con la solución propuesta se muestra cómo un programa genérico (reutilizable) en el PLC se adapta a diferentes casos específicos de aplicación sin requerir modificaciones importantes en el código.

El presente documento está organizado de la siguiente manera: en la sección II se muestra un marco de referencia del área del proyecto. La Sección III se muestra la aplicación de iMRP en la abstracción de particularidades en sistemas basados en PLC. En la sección IV se muestra a manera de validación del esquema propuesto la aplicación del sistema en varios casos específicos. Finalmente se describen las conclusiones de mayor relevancia obtenidas en el desarrollo del proyecto.

1. MARCO TEÓRICO

1.1. Operación dirigida por modelo.

El concepto de operación dirigida por modelo propuesto en [28] para el control del conjunto de equipos de fabricación de un sistema de fabricación, busca facilitar la aplicación de un sistema informático (genérico) en el control de diferentes casos específicos, sin modificaciones sustantivas en el código. Su aplicación en el control basado en PLC es reportado en [Electro 202 Mario], mediante el control de secuencia de operación de diferentes tipos de sistemas, utilizando el mismo sistema de control (genérico).

1.2. Operación dirigida por modelo en sistemas basados en PLC.

La técnica iMRP es adaptada al modelado de sistemas basados en PLC como se describe en [tesis] y [artículo electro mario]. En el caso de sistemas basados en PLC el subgrafo producto se propone para representar la secuenciación de fases de operación del sistema. Por otra parte el subgrafo que representa a los equipos, puede adaptarse para representar elementos actuadores del sistema. El subgrafo recurso tiempo permite representar los pasos de operación.

Sistema Genérico (modelo de referencia).

La aplicación del concepto operación dirigida por modelo implica tres aspectos: representación del modelo de particularidades del sistema, lectura del modelo para instanciación de elementos y lectura del modelo para operación del sistema, Figura 1.

Representación del modelo de particularidades. El modelo gráfico de particularidades del caso específico de control debe estar implementado de manera (o forma) disponible al sistema genérico operando en el PLC. Para tal efecto el modelo gráfico puede ser representado en forma de un conjunto de matrices, con base en (1).

$$M(r, c) = 1 \text{ si nodo } c \text{ es padre de nodo } r \quad (1)$$

$$= 0 \text{ de otra forma (otherwise).}$$

Instanciación de los elementos del caso específico. En IEC 61131-3 standard las clases son implementadas en function blocks, los términos clase y function block se pueden manejar como sinónimos, [citar PLC object-oriented programming]. Aun y cuando los function blocks pueden implementarse en lenguajes del estándar IEC 61131-3, la definición del tipo y la instanciación se hace en tiempo de diseño. Tal característica de los function block implica rigidez a un sistema genérico debido a que el uso de un sistema genérico debe ser independiente del número y tipo de objetos requeridos por el caso específico. En el caso de operación dirigida por modelo, el modelo de particularidades define al sistema genérico los objetos de cada caso específico de aplicación para su instanciación a partir de clases que han sido implementadas en el sistema genérico o bien para insertar en el código del propio sistema genérico en tiempo de ejecución.

El sistema genérico cuenta con bloques funcionales (clases) implementados para cada tipo de nodo presentes en el modelo de particularidades (Fase, RT y Actuador). El sistema genérico

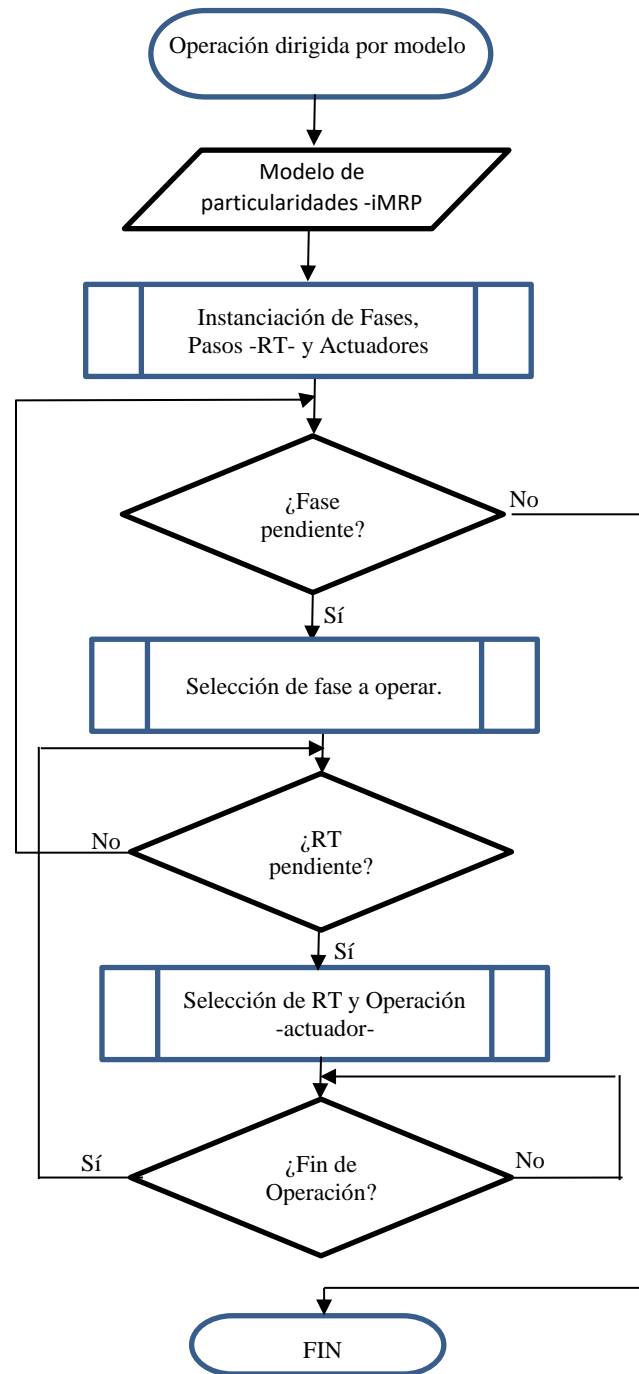


Figura 1. Sistema de Aplicación Genérica.

está compuesto por un núcleo o elemento principal, y tres tipos de bloques funcionales.. El núcleo contiene los arreglos arrFase, arrRT y arrActuador, para el manejo de los n elementos o instancias (nodos) existentes en un caso específico (modelo de particularidades), Figura mmm. En todos los elementos

existen los atributos Var1 y Var2, a utilizar como variables comodín según sea requerido por el caso (dominio) de aplicación específico.

El sistema genérico con base en las matrices del modelo de particularidades, inicializa cada uno de los elementos en los arreglos de subgrafo (arrFase, arrRT, arrActuador). Este proceso de inicialización incluye asignar a cada objeto CFase los objetos CRTs correspondientes, así mismo, asignar a cada objeto CRT su correspondiente objeto CActuador. De la misma manera incluye la lectura de los atributos de nodo y arco, en particular de los atributos Var1 y Var2.

Los objetos correspondientes a los nodos de un subgrafo Fase o RecursoTiempo tendrían el mismo código (mismo tipo o function block. Por otra parte los objetos de subgrafo Actuador o Equipo requieren realizar diferente función. La solución propuesta consiste en implementar en un solo block funcional (o tipo) el código requerido para la operación de cada uno de los actuadores. La sección de código de la rutina operar a ejecutar dependerá del actuador asociado al paso correspondiente.

Operación del Sistema. El programa identifica los elementos CFase en condición de ser operados, esto es, nodos sin nodos hijos en el subgrafo Fase, o bien nodos ya operados. Los objetos correspondientes a tales nodos (elementos en arrFase) son llamados a operar a través de su rutina Operar(). Esta rutina llama a operar el objeto RT correspondiente y este a su vez a su objeto actuador. Es importante destacar que los objetos nodo actuador implementan el manejo de los distintos actuadores/sensores conectados al PLC. En cada fin de operación el sistema revisa la matriz fase (actualizada) para identificar nodos que se encuentren con la totalidad de nodos – Fase- hijos que hayan finalizado su operación. De esta manera, el subgrafo Fase indica al sistema genérico la secuencia (condiciones de precedencia) para la realización de cada una de las fases del caso específico de aplicación del sistema.

La realización de cada fase consiste de uno o varios pasos representados cada uno por nodos del subgrafo RT. El objeto CFase a operar identifica el paso a realizar con base a sus RTs asociados de acuerdo al modelo (matriz Fase-RT) y envía a operar al objeto del arreglo arrRT correspondiente al paso (objeto RT) a ejecutar. Cada uno de los pasos (nodos RT) del modelo se encuentran asociados a un objeto actuador (nodo Actuador). El objeto RT en ejecución envía a operar a su objeto actuador correspondiente. La clase del objeto actuador implementa la rutina operar de acuerdo a la funcionalidad a realizar por el actuador. Dado que cada objeto actuador requiere una funcionalidad diferente se requiere diferente implementación de la rutina operar en cada objeto actuador. Ante la falta de capacidad para hacer enlace tardío (late binding), tal referencia debe especificarse en el código en tiempo de diseño del sistema por lo que se propone el uso de un solo tipo de bloque funcional actuador. En la rutina operar de

este tipo de actuador se implementa el código correspondiente a cada rutina operar requerida en cada uno de los objetos actuadores. Así el objeto Actuador recibe el ID del paso (objeto RT) a realizar., y ejecuta sólo el código correspondiente al paso a realizar (activa actuador(es) y lee sensor(es) asociados al paso para indicar el fin de operación.

La funcionalidad del tipo RT (paso) se encuentra implementada básicamente en dos rutinas: AgregarActuador y Operar y FinDeOperacion. La rutina AgregarActuador es utilizada en el proceso de inicialización para asociar al objeto RT el objeto actuador especificado en el modelo de particularidades (RT-Actuador). Su rutina Operar envía a operar al objeto actuador asociado al paso.

De esta manera el bloque funcional que depende del caso específico es el bloque funcional Actuador. Por lo que el cambio de caso específico de aplicación requiere únicamente cambio en el tipo de bloque funcional Actuador. Como elemento de soporte para mayor flexibilidad del sistema se encuentran los atributos Var1 y Var2 en los bloques funcionales que soportan la adaptación a diferentes condiciones como pueden ser iteraciones a realizar, tiempos, entre otros.

2. DESARROLLO.

2.1. Modelado de Particularidades.

En la presente sección se describe la aplicación del concepto operación dirigida por modelo en el caso del control de una célula robotizada controlada por PLC. La operación de la célula consiste en el Robot que toma una pieza en la entrada de la célula y la coloca para su procesamiento en el Centro de Maquinado. Posteriormente el Robot toma la pieza del Centro de Maquinado para colocarla en la salida de la célula. El modelo de particularidades se muestra en la figura 2. La secuencia de operación de los equipos está definida en dos partes. Una parte está dada por la relación nodo hijo-nodo padre del subgrafo Fase (lógica iMRP): Fase C - Fase B. La otra parte corresponde a la relación nodo padre-nodo hijo del subgrafo Recurso Tiempo; ejemplo el caso de fase C, asociada a los dos equipos, el nodo padre RT2, asociado al Robot R, indica operación del Robot R. Enseguida el nodo hijo RT3 asociado al Centro de Maquinado CNC, indica operación del Centro de Maquinado CNC.

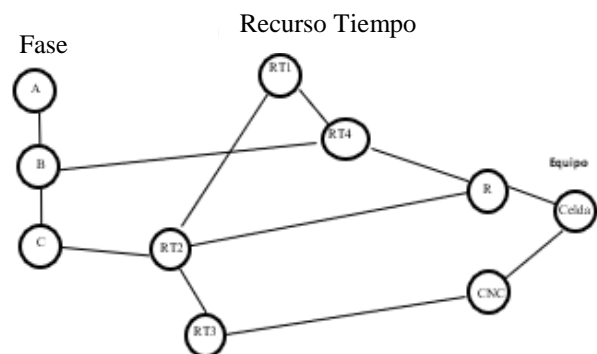


Figura 2. Modelo de particularidades del Sistema -iMRP-.



Figura 3. Célula Robotizada de Fabricación.

La característica principal del programa genérico es su capacidad para interpretar y enviar a operar cada uno de los casos específicos de aplicación con base en el modelo de particularidades correspondiente.

En el controlador de los equipos (CNC y Robot) se encuentran disponibles entradas/salidas que permiten la puesta en marcha de los mismos de manera remota (desde el PLC). En la figura 3 se muestra una selección de las mismas a utilizar en la descripción de la operación de la célula. En ambos equipos (R y CNC) se han cargado los programas a ejecutar en la fase correspondiente.

En el PLC se encuentra el programa genérico que incluye el modelo de particularidades de la célula. Mediante entradas y salidas del PLC se inicia la operación del programa en el equipo de fabricación respectivo. En el caso del Robot, se inicia su operación mediante su entrada Inicio, así como la selección del programa a ejecutar (Entradas: Prog1, Prog2, Prog3). La salida del Robot Fin ciclo, señala al PLC el fin de ejecución del programa. Por otra parte, de manera similar el centro de maquinado CNC es comandado mediante sus entradas (salidas del PLC). En el momento de operar uno de los equipos, el objeto actuador ejecuta la rutina necesaria para activar/desactivar las señales correspondientes para la operación del equipo. Tal rutina no requiere incluir la lógica de operación de la célula simplificando así el código de la misma. Así, debido a que la lógica de operación de la célula se encuentra representada en forma gráfica en el modelo, la rutina de operación de cada equipo (actuador) se simplifica considerablemente.

2.2. Operación del Sistema

La primera fase a ejecutar en el sistema es la fase C asociada a los nodos Recurso Tiempo RT2 y RT3, asociados a su vez a los nodos R y CNC, respectivamente, ver FIGURA 2. Posteriormente se ejecuta la fase B asociada al nodo Recurso Tiempo RT4, asociado a su vez al nodo R, como se ilustra a continuación:

- 1.- C-RT2-R 2.-C-RT3-CNC 3.-B-RT4-R

A continuación, se muestra la funcionalidad de la rutina operar para cada bloque funcional correspondiente a cada paso de cada uno de los equipos -Nodos Actuador-:

- Paso 1 -Robot Operar()**
 Activar salida Abrir Puerta
 Revisar entrada Puerta Abierta
 Activar salida Prog1
 Activar salida Inicio
- Paso 2 CNC Operar()**
 Activar salida Cerrar Puerta
 Revisar entrada Puerta Cerrada
 Activar salida Inicio
- Paso 3 Robot Operar()**
 Activar salida Abrir Puerta
 Revisar entrada Puerta Abierta
 Activar salida Prog2
 Activar salida Inicio

De esta manera se facilita el desarrollo del programa de control de la célula ya sea para un nuevo caso de aplicación o bien para una actualización en el conjunto de equipos de la célula. En ambos casos el proceso de desarrollo del sistema de control se

enfoca a la actualización del modelo (gráfico) de particularidades y en la escritura de la rutina operar de los equipos (o del nuevo equipo).

3. CONCLUSIONES.

EL concepto de operación dirigida por modelo facilita la aplicación de un sistema genérico en casos específicos o particulares. Es posible la integración e interpretación del modelo de naturaleza gráfica iMRP en el programa genérico del PLC, mediante matrices y bloques funcionales. La separación del sistema particular en sistema genérico y modelo de particularidades, facilita el desarrollo de sistemas de control mediante la reusabilidad del sistema genérico en la interpretación de distintos modelos de particularidades.

Referencias

- [1] Naur, P., Randell, B. Eds. Software Engineering Conf. NATO Science Committee, Garmisch, Germany, Oct, 1968.
- [2] Goldstine, H.H. and von Neumann, J. Planning and coding of problems for an electronic computing instrument, Part II, Volume 1. Report prepared for US Army Ordnance Department, April 1947.
- [3] Jilani, A. A.A., Ussman, M., Haim, Z. Model Transformation in Model Driven Architecture. Universal Journal of Computer Science and Engineering Technology. Vol 1, No.1, Oct. 2010.
- [4] Kahani, N., Bagherzadeh, M., Cordy, J.R., Dingel, J., Varro, D. Survey and Classification of model transformation tools. Software and System Modeling(. Marzo 2018.
- [5] Fabra, J., De Castro, V., Álvarez, P., Marcos, E. Automatic Execution of Business Process Model: Exploiting the Benefits of Model-Driven Engineering Approaches. The Journal of Systems and Software. Vol.85.pp. 607-625. 2012
- [6] Object Management Group. Model Driven Architecture (MDA) MDA Guide Rev. 2.0. OMG Document ormsc/2014-06-01.
- [7] Rodrigues da Silva, A. Model-driven engineering: A survey supported by the unified conceptual model. Computer Languages, Systems & Structures. Vol. 43, pp. 139-155. 2015.
- [8] Muller, G., Korfiatis, P. Eds. Architecture Deployment. White Paper Resulting from Architecture Forum Meeting. November, 2010. Philips Healthcare, Best, The Netherlands.
- [9] Coupat, R., Philippot, A., Niang, M., Riera, B. et al. Methodology for Railway Automation Study and Automatic Generation of PLC Programs. IEEE Intelligent Transportation System Magazine. Vol. 80. Fall 2018.
- [10] Sunitha, E.V., Samuel, P. Automatic Code Generation From UML State Chart Diagrams. IEEE Access. Vol 7. 2019.
- [11] Ciccozzi, F., Malavolta, I., Selic, B. Execution of UML models: a systematic review of research and practice. Software and system modeling. Vol. 18. Pp. 2313-2360. 2019.
- [12] Xaver, R., Hubber, R., Christian, L., Rogliner, M. Capturing Smart service systems. Development of a domain specific modelling language. J. of Information Systems. Vol. 29. 2019. pp. 1207-1255.
- [13] Yousaf, N. Ahmad, A., Hamdani, M., Ashfaq, M., Siddiqui, H. Investigation of Modern Tools and Techniques Provided by Modeling Editors: A Survey. 7th Conf. On Computer and Communications Management. pp. 96-. 2019
- [14] OMG. OMG System Modeling Language (OMG-SysML). Version 1.6. OMG document No. Formal/19-11-01.
- [15] OMG. (2019). UML Profile for MARTE : Modeling and Analysis of Real-Time Embedded Systems v.1.0. In *An OMG UML Profile for Marte* (Issue Abril). <http://www.omg.org/spec/MARTE/1.1/>
- [16] Thramboulidis, K., Christoulakis, F., UML4IoT – A UML based approach to exploit IoT in Cyber Physical manufacturing systems. Computers in Industry. 2016.
- [17] Bendraou, R., Gervais, M.-P., & Blanc, X. (2005). UML4SPM: A UML2 .0-Based Metamodel for Software. *Structure*, 6(511731), 17–38. http://dx.doi.org/10.1007/11557432_3
- [18] Barbieri, G., Battilani, N., & Fantuzzi, C. (2015). *PackML-based Design Pattern Design Pattern Design Pattern Design Pattern Modular PLC Code Modular PLC Code Modular PLC Code Modular PLC Code*. 10, 178–183. <https://doi.org/10.1016/j.ifacol.2015.08.128>
- [19] Witsch, D., & Vogel-Heuser, B. (2011). PLC-statecharts: An approach to integrate UML-statecharts in open-loop control engineering - Aspects on behavioral semantics and model-checking. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 44(1 PART 1), 7866–7872. <https://doi.org/10.3182/20110828-6-IT-1002.02207>
- [20] Wang, W., Member, S., Niu, N., & Member, S. (2019). In-Place Traceability for Automated Production Systems : A Survey of PLC and SysML Tools. *IEEE Transactions on Industrial Informatics*, 15(6), 3155–3162. <https://doi.org/10.1109/TII.2018.2878782>
- [21] Darvas, D., Viñuela, E. B., & Majzik, I. PLC Code Generation Based on a Formal Specification Language. *14th IEEE International Conference on Industrial Informatics (INDIN)*, 389–396. 2016
- [22] IEC-60848. (2013). *GRAFCET specification language for sequential function charts, IEC Std., 2013.*,
- [23] ADEPA. (1981). *GEMMA (Guide d'Étude des Modes de Marches et d'Arrêts)*.
- [24] Wortmann, J. C., Hegge, H. M. H., Goossenaerts, J. B. M. Understanding Enterprise Modelling From Product Modelling. *Production Planning & Control*. 2001. 12(3), 234–244.,
- [25] Geiger, M., Harrer, S., Lenhard, J., & Wirtz, G. (2018). BPM 2.0: The state of support and implementation. *Future Generation Computer Systems*, 80. <https://doi.org/10.1016/j.future.2017.01.006>
- [26] Lijima, K., Natis, Y., Driver, M., Vincent, P., & Wong, J. (2019). Gartner Magic Quadrant for Enterprise Low-Code Application Platforms. Report ID G00361584 (Issue August). Gartner Inc.
- [27] Acosta-Cano-de-los-Rios, J. (2016). Esquema de Referencia para Acoplamiento Débil entre Sistema Informático y Equipo de Producción. Universidad Politécnica de Madrid